TIPP: taxonomic identification and phylogenetic profiling

Nam-phuong Nguyen¹, Siavash Mirarab¹, Bo Liu², Mihai Pop² and Tandy Warnow¹,³,⁴,*
¹Department of Computer Science, The University of Texas at Austin, Austin, TX USA, ²Center for Bioinformatics and Computational Biology, University of Maryland at College Park, College Park, MD USA, ³Department of Bioengineering, The University of Illinois at Urbana-Champaign, Urbana, IL USA and ⁴Department of Computer Science, The University of Illinois at Urbana-Champaign, Urbana, IL USA

Associate Editor: Gunnar Ratsch

ABSTRACT

Motivation: Abundance profiling (also called ‘phylogenetic profiling’) is a crucial step in understanding the diversity of a metagenomic sample, and one of the basic techniques used for this is taxonomic identification of the metagenomic reads.

Results: We present taxon identification and phylogenetic profiling (TIPP), a new marker-based taxon identification and abundance profiling method. TIPP combines SAT*e-enabled phylogenetic placement a phylogenetic placement method, with statistical techniques to control the classification precision and recall, and results in improved abundance profiles. TIPP is highly accurate even in the presence of high indel errors and novel genomes, and matches or improves on previous approaches, including NBC, mOTU, PhymmBL, MetaPhyler and MetaPhlAn.


Contact: warnow@illinois.edu

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on April 5, 2014; revised on October 12, 2014; accepted on October 24, 2014

1 INTRODUCTION

Metagenomic studies of microbial communities commonly generate millions to hundreds of millions of sequencing reads. The assignment of accurate taxonomic labels to these sequences is a critical component in many analyses, but is complicated by the fact that the majority of the organisms found in environmental or host-associated communities cannot be easily cultured in a laboratory, and of the organisms that can be cultured, an even smaller number have been sequenced, even partially. Thus, these commonly encountered organisms are largely absent from existing databases of known genomes and genes. Providing taxonomic labels to metagenomic sequences requires extrapolating the knowledge contained in sequence databases to previously unseen deoxyribonucleic acid (DNA) strings. Simple similarity-based approaches (e.g. picking the best database hit as the best ‘guess’ at the taxonomic label) have been shown to be insufficiently accurate (Koski and Golding, 2001), leading to the development of new and more sophisticated methods.

Recently developed methods improving on the simple similarity-based approaches include (a) composition-based approaches that rely on various machine learning techniques (Support Vector Machines in PhyloPythia and PhyloPythiaS (McHardy et al., 2007; Patil et al., 2011), Interpolated Markov Models in Phymm (Brady and Salzberg, 2011), Bayesian models in NBC (Rosen et al., 2011), or neural networks (Abe et al., 2006)) to classify sequences based on their DNA composition (usually based on the frequency of short k-mers), (b) more sophisticated analyses of similarity search results [e.g. using lowest common ancestor aggregation in Megan (Huson et al., 2007), or classifiers built from similarity search results as done in MetaPhyler (Liu et al., 2010, 2011), MetaPhiAn (Segata et al., 2012) and mOTU (Sunagawa et al., 2013) or protein profiles in Carma (Gerlach and Stoye, 2011)], and (c) combinations of multiple approaches (e.g. composition and similarity based approaches in PhymmBL (Brady and Salzberg, 2009)). Some of these approaches (e.g. most of the composition-based approaches) can be applied to any DNA sequence. However, others are specific to a reference collection of carefully selected genes (e.g. MetaPhyler, MetaPhiAn and mOTU use a collection of universal or clade-specific marker genes), and so are called ‘marker-based’ methods.

Abundance profiling, also called ‘phylogenetic profiling’, seeks to estimate the relative abundance of the species (or genera, or families, etc.) within a sequence dataset. While many methods produce these estimates by characterizing most (or all) of the sequences in the dataset, marker-based methods produce these estimates by characterizing only those sequences that match the marker genes they rely on. The marker genes used by some of these methods (e.g. MetaPhyler and mOTU) are supposed to be single copy and universal; thus, the estimates produced using such markers can be used for profiling without needing to correct for copy number or missing data.

We present taxon identification and phylogenetic profiling (TIPP), a new marker-based method for taxon identification and abundance profiling. We explore TIPP for abundance profiling using MetaPhyler’s collection of 30 marker genes, but TIPP can be used with any single-copy gene for which a dense enough sample of full-length sequences is available. TIPP is implemented in Python and can be run on Unix and Mac.

2 APPROACH

2.1 Overview

TIPP has three phases: pre-processing (Phase 1), taxon identification for reads mapped to its marker genes (Phase 2) and abundance profiling based on these taxon identifications (Phase 3).
Phase 1 involves a sequence of preprocessing steps, based on its set \( S \) of reference sequences for its set of marker genes, and its input set \( Q \) of uncharacterized reads (the ‘query sequences’). TIPP uses SATé (Liu et al., 2009, 2012) to compute an alignment \( A \) and tree \( T \) for full-length sequences for each of its marker genes, using its reference set \( S \); these are called the ‘backbone alignment’ and ‘backbone tree’. Since the NCBI taxonomy is not fully resolved (i.e., bifurcating), TIPP uses the backbone alignment and RAxML (Stamatakis, 2006) to refine the NCBI taxonomy (restricted to the sequences in the backbone alignment) into a fully resolved tree, \( T^* \). Thus, each marker gene has its own refinement of the taxonomy that is used in subsequent analyses. TIPP then uses BLAST (Altschul et al., 1990) to determine which reads in \( Q \) map to its marker genes. At the end of this pre-processing, the set \( Q \) is partitioned into sets, with one set for each of its marker genes, and a final set for the reads that are not mapped to any marker gene.

Phase 2, where reads that are mapped to marker genes are taxonomically characterized, is the most involved, and is described below (and also in Fig. 1).

Phase 3, which estimates the abundance profile, is quite simple. We pool all the reads that mapped to any marker gene and have been characterized at any taxonomic level into a single set. We compute the abundance profile as the relative distribution of the clades present in the pooled set. As the pooled reads are only required to be characterized at the phylum level, the abundance profiles can include a category of unclassified sequences. Sequences can be labeled as unclassified due to two reasons: there may not be sufficient support to classify a sequence at a given taxonomic level, and the taxonomy may not have a classification for a clade at a particular taxonomic level. For example, a species that does not have a family label would be unclassified at the family level, even if it was classified at a lower level; see Section S7 for an example.

2.2 Phase 2: taxonomic characterizations of reads

We describe how TIPP performs its taxonomic characterization of a single read mapped to a single marker gene; see Figure 1. The input to TIPP includes the maximum alignment subset size \( m \) and the statistical support thresholds \( s_q \) and \( s_p \) for alignment and placement support, respectively, which can be set by the user. We set default values as follows: \( m = 100 \) and \( s_q = s_p = 95\% \).

Step 1: Decomposition. TIPP decomposes the set of leaves in the backbone tree \( T \) into subsets using the decomposition technique in SEPP. TIPP finds a centroid edge in \( T \) (one that separates the leaf set into two sets of approximately equal size), breaks the tree accordingly into two subtrees, and recurses on each subtree that contains more than \( m \) leaves. This produces a partition of the leaves of \( T \) into alignment subsets \( S_1, S_2, \ldots, S_k \), each of size at most \( m \).

Step 2: Compute Extended Alignments. We define the alignment \( A_i \) by restricting the alignment \( A \) to \( S_i \), for every \( 1 \leq i \leq k \); these are the ‘subset alignments’. TIPP uses the HMMER software suite (Eddy, 1998) to compute an HMM \( H_i \) on each \( A_i \). Thus, TIPP represents the backbone alignment with a set of HMMs, a technique we call an ‘HMM Family’. HMMER produces bit scores (discussed below), which are measures of the fit between each \( H_i \) and each query sequence \( q \). Then, for the query sequence \( q \), one or more subset alignments are selected so that the total statistical support for the alignments is at least \( s_q \) (see below for how TIPP computes statistical support). TIPP uses HMMER to align \( q \) to each of the selected subset alignments, and thus, produces extensions of each subset alignment \( A_i \) that include \( q \) (called an ‘extended alignment’). Thus, each query sequence \( q \) gives rise to at least one and potentially many extended alignments of the reference dataset, each with \( |S| + 1 \) sequences.

Step 3: Placement. For each query sequence \( q \) and each extended alignment containing \( q \), we use pplacer (Matsen et al., 2010), a maximum likelihood phylogenetic placement method, to insert \( q \) into \( T^* \). The output of pplacer provides multiple placements and their likelihood weight ratios (the maximum likelihood values for all placement locations, normalized to sum to 1) for each extended alignment. We combine all placements resulting from the extended alignments into a single collection of placements, re-normalize their likelihood weight ratios to sum up to 1 (weighting each placement by the corresponding alignment support), and treat the normalized likelihood weight ratios as probabilities; see Section S8 for an example.

Step 4: Classification. For each query sequence, we assign statistical support to each node in the taxonomy by adding the probabilities of all placements at or below the edge above the node; thus, the statistical support monotonically increases as one traverses the tree from any leaf to the root, and this allows us to classify the query sequence at all taxonomic levels for which it has support of at least \( s_p \). The query sequence is left unclassified at levels where the support of \( s_p \) is not reached. A unique taxon identification is produced whenever \( s_p > 0.5 \); otherwise, TIPP outputs all identifications that meet the support threshold of \( s_p \), along with their support values. Because our default setting has \( s_p = 0.95 \), TIPP produces a unique taxon identification for each query sequence.
2.3 Alignment support calculation

To take alignment uncertainty into account, we take a large enough number of extended alignments to reach the alignment support threshold $s_a$. To determine the number of extended alignments we need, we use HMMER’s output to define the probability that a given query sequence is generated by a given HMM from the set of HMMs computed for the different subset alignments. These calculations are based on the assumptions that (1) the subsets are disjoint, so that at most one HMM generates the query sequence and (2) the query sequence is generated by some HMM.

For a given HMMER model $H$ and query sequence $q$, HMMER calculates a bit-score (which we denote $\text{BS}(H)$), defined by:

$$\text{BS}(H) = \log \frac{P(q|H)}{P(q|R)}$$

(1)

where $P(q|H)$ is the probability of model $H$ generating query sequence $q$, and $P(q|R)$ is the probability of a random model $R$ generating query sequence $q$. Assuming that the query sequence $q$ is generated by exactly one of the HMMs ($H_1$ to $H_m$ each corresponding to a different subset alignment), the probability that $H_i$ generated $q$ is:

$$P(H_i|q) = \frac{P(q|H_i)P(H_i)}{\sum_{j=1}^{m} P(q|H_j)P(H_j)}.$$  

(2)

Since our subsets all have roughly equal size, we make the simplifying assumption that the a priori probabilities of any two HMMs generating any given query sequence are equal. We can rewrite Equation (2) as:

$$P(H_i|q) = \frac{1}{\sum_{j=1}^{m} P(q|H_j)P(H_j)}.$$  

(3)

By Equation (1),

$$\text{BS}(H_i) = \text{BS}(H) - \text{BS}(H_j) = \log \frac{P(q|H_i)}{P(q|R)} - \log \frac{P(q|H_j)}{P(q|R)} = \log \frac{P(q|H_i)}{P(q|H_j)}.$$  

(4)

(5)

Hence, the probability of $H_i$ using bit-scores is given by:

$$P(H_i|q) = \frac{1}{\sum_{j=1}^{m} \text{BS}(H_i) - \text{BS}(H_j)}.$$  

(6)

Thus, assuming that the bit-scores are sorted such that $\text{BS}(H_i) \geq \text{BS}(H_{i-1})$ ($i = 1, 2, \ldots, n - 1$), to reach a specified threshold $s_a$, we find the smallest $m$ such that $\sum_{k=1}^{m} P(H_k|q) \geq s_a$.

3 EXPERIMENTAL DESIGN

3.1 Overview

We evaluated TIPP in comparison to other phylogenetic profiling methods under three different conditions. Experiment 1 compared performance under easy conditions, where the genomes are known and the reads (i.e. query sequences) have low rates of sequencing errors. Experiment 2 examined datasets with high rates of insertion and deletion errors (collectively known as ‘indels’). Experiment 3 examined performance where the query sequences come from ‘novel’ genomes (defined below).

3.1.1 Methods studied We compared TIPP, MetaPhyler, MetaPhlAn, PhymmBL, mOTU and NBC. Three of these methods (TIPP, mOTU and MetaPhyler) are marker-based methods, and use universal housekeeping genes that are unlikely to undergo duplication or horizontal gene transfer. Both TIPP and MetaPhyler use the same collection of 30 marker genes, and mOTU uses a reduced set of 10 marker genes selected from Mendes et al. (2013). MetaPhlAn, conversely, selects markers that uniquely identify specific taxonomic groups. Assigning query sequences to genes (i.e. binning) is performed internally by each of the marker-based methods; see Section S4 for the BLAST settings used within TIPP for binning.

Phred quality scores (Ewing and Green, 1998) are needed for each read, in order for mOTU to run. Since the datasets used in our study do not have the quality scores, we assigned Phred quality scores of 33 to all the bases of all the reads (99.95% probability that the base is correct). MetaPhyler allows the user to input a confidence threshold, and query sequences are only classified at a given level if this confidence threshold is met. We use the confidence level of 90% suggested by the authors of MetaPhyler. For PhymmBL, we classify a query sequence at the most specific classification yielding a confidence score of 95% or higher; however, PhymmBL does not give confidence scores at the species level, and thus, cannot be used to perform abundance profiling at the species level. Finally, NBC gives a confidence score of the query sequence matching a taxon. We accept the classification if the confidence score is above the species threshold formula given by the NBC authors (which is a function of read-length; see Section S4); thus, a query sequence will either be classified at the species level or be completely unclassified. See Section S4 for version numbers and commands used.

3.1.2 Reference marker datasets TIPP uses the reference sequence dataset obtained from Liu et al. (2010, 2011), consisting of 30 phylogenetic marker genes that span the Bacteria and Archaea domains. The marker genes selected were believed to be single copy genes, universally present across the Bacteria domain, and resistant to horizontal gene transfer. Only species whose genomes have been sequenced were present in the reference dataset. The number of sequenced representatives for each marker gene ranges from 65 to 1555 sequences, with an average of 1312 sequences per marker gene. See Section S2 for the list of marker genes and the empirical statistics of the reference alignments on these datasets.

3.1.3 Training MetaPhyler, MetaPhlAn and mOTU come pre-trained, but their reference datasets depend on which version is used. NBC and PhymmBL require the user to manually train their software, and their reference datasets are dependent on when the genomes were downloaded for training. The versions of MetaPhyler, MetaPhlAn and mOTU used in our article are based on reference datasets downloaded from NCBI in 2010, 2011 and 2012, respectively. TIPP is trained on the same reference dataset as MetaPhyler, and thus is trained on sequences downloaded from NCBI in 2010. NBC and PhymmBL were
trained on the same set of genomes downloaded from NCBI on May 2013. Thus, NBC and PhymmBL are based on training datasets from 2013, mOTU is based on a training dataset from 2012, MetaPhlAn is based on a training dataset from 2011, and MetaPhyler and TIPP are based on a training dataset from 2010. These differences favour methods (such as NBC and PhymmBL) that are trained on more recent datasets than those based on older datasets (such as MetaPhyler and TIPP). See Section S6 for additional information on how TIPP was trained.

3.1.4 Simulated abundance profiling datasets The datasets we used have different properties, including the average read length (short versus long), complexity of the profiles (uniform versus non-uniform), the rate of sequencing errors (low versus high), and whether the datasets contained ‘novel’ or ‘known’ genomes (defined shortly). The complexity of the profile is labeled as low complexity (LC; staggered distribution of species), high complexity (HC; uniform distribution of species) and medium complexity (MC; distribution of species in between LC and HC). A dataset is labeled as ‘novel’ if none of the marker-based methods have been trained on any of the genomes in that dataset (see below for more detail). A dataset is labeled as ‘known’ if it contains any genome that was previously used to train at least one of the marker-based methods. We provide a brief overview of the datasets, separated out into experimental conditions (Table 1); a more in-depth description of the individual datasets (including their Shannon Entropy) can be found in Section S3.

3.1.5 Experimental condition 1: easy datasets The first group of datasets contained query sequences from known genomes (all the genomes were present in the training sets of at least one of the marker-based methods) and had low rates of sequencing error (insertions, deletions and substitutions). The easy datasets are separated into two different conditions: long read datasets (average length of 200 to 1000 nucleotides) and short read datasets (average length of 100 nucleotides or shorter). The long read datasets include the FACS HC simulated dataset (Stranneheim et al., 2010), the FAMEs LC, MC and HC datasets (Mavromatis et al., 2007), and the WebCarma HC-simulated dataset (Gerlach and Stoye, 2011). Both the WebCarma and FACS datasets contained simulated 454 reads and had an average indel rate of 3% per base. The FAMEs dataset contained reads taken from Sanger sequencing projects, and thus, is expected to have very low rates of sequencing error [typically <1% error per base (Shendure and Ji, 2008)]. The original FACS HC dataset contained viral, bacterial, and human sequences, and we removed the viral and human sequences so that profiles were estimated only on bacterial reads. The short read datasets include the MetaPhlAn HC and LC-simulated datasets (Segata et al., 2012), the TIPP FACS HC Illumina-simulated dataset, and the TIPPWebCarma HC Illumina-simulated dataset; the last two datasets were generated using MetaSim (Richter et al., 2008) to simulate Illumina reads from the metagenomes used in the original FACS HC and WebCarma HC datasets. The short read datasets contain simulated Illumina reads with at most 0.2% substitutions per base.

3.1.6 Experimental condition 2: datasets with high rates of indel errors The second group of datasets was modified versions of the easy datasets with additional indels to make analyses more difficult. The easy datasets contained reads with less than 0.2% substitution errors per base or less than 3% indel errors per base. We inserted additional indel sequencing errors to the query sequences in the easy datasets, using indel rates typical of PACBio reads (12% insertion and 2% deletion error rate per base (Carneiro et al., 2012)). Details can be found in Section S3.6.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Experiment</th>
<th># Genomes</th>
<th>Complexity</th>
<th># Reads</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaPhlAn HC</td>
<td>1&amp;2</td>
<td>100</td>
<td>High</td>
<td>1000000</td>
<td>88 (s)</td>
</tr>
<tr>
<td>MetaPhlAn LC</td>
<td>1&amp;2</td>
<td>25</td>
<td>Low</td>
<td>240000</td>
<td>88 (s)</td>
</tr>
<tr>
<td>FAMEs HC</td>
<td>1&amp;2</td>
<td>113</td>
<td>High</td>
<td>116771</td>
<td>949 (l)</td>
</tr>
<tr>
<td>FAMEs MC</td>
<td>1&amp;2</td>
<td>113</td>
<td>Medium</td>
<td>114457</td>
<td>969 (l)</td>
</tr>
<tr>
<td>FAMEs LC</td>
<td>1&amp;2</td>
<td>113</td>
<td>Low</td>
<td>97495</td>
<td>951 (l)</td>
</tr>
<tr>
<td>FACS HC-454</td>
<td>1&amp;2</td>
<td>19</td>
<td>High</td>
<td>26894</td>
<td>268 (l)</td>
</tr>
<tr>
<td>TIPP FACS HC-Illum</td>
<td>1&amp;2</td>
<td>19</td>
<td>High</td>
<td>300000</td>
<td>100 (s)</td>
</tr>
<tr>
<td>WebCarma-454</td>
<td>1&amp;2</td>
<td>25</td>
<td>High</td>
<td>25000</td>
<td>265 (l)</td>
</tr>
<tr>
<td>TIPP WebCarma-Illum</td>
<td>1&amp;2</td>
<td>25</td>
<td>High</td>
<td>300000</td>
<td>100 (s)</td>
</tr>
<tr>
<td>TIPP HC novel-Illum</td>
<td>3</td>
<td>100</td>
<td>High</td>
<td>1000000</td>
<td>100 (s)</td>
</tr>
<tr>
<td>TIPP LC novel-Illum</td>
<td>3</td>
<td>100</td>
<td>Low</td>
<td>1000000</td>
<td>100 (s)</td>
</tr>
<tr>
<td>TIPP HC novel-454</td>
<td>3</td>
<td>100</td>
<td>High</td>
<td>1000000</td>
<td>269 (l)</td>
</tr>
<tr>
<td>TIPP LC novel-454</td>
<td>3</td>
<td>100</td>
<td>Low</td>
<td>1000000</td>
<td>269 (l)</td>
</tr>
</tbody>
</table>

Eleven basic datasets were used in this study, but each dataset has two versions – a low error model (with mostly substitutions and few or no indels) and a high error model (where the indel rate is high); see text for details about the error models for all datasets. Datasets labelled ‘TIPP’ were generated for this study; the rest are datasets previously studied in the literature. ‘Experiment’ specifies the experiments where each dataset is used. The low indel versions are studied in Experiments 1 and 3, and the high indel versions are studied in Experiments 2 and 3. The number of genomes in the dataset is given in the column labelled ‘# Genomes’. The complexity of the dataset is given in the column labelled ‘Complexity’. The number of reads in the dataset is given in the column labelled ‘# Reads’. ‘Length’ refers to the average length of the reads for the low indel version of the dataset (‘s’ and ‘l’ refer to short and long).
3.1.7 Experimental condition 3: datasets with novel genomes. We generated datasets that contained ‘novel’ genomes, which are genomes that were not in the training sets of the marker-based methods. As the marker-based methods use older versions of the NCBI genomes compared to the composition-based methods, some of the selected genomes were present in the training set of the composition-based methods. To make the genomes novel to all methods, we modified the training sets of the composition-based methods to exclude the selected genomes. This modification was only applied to Experiment 3, where we analyzed the novel genome datasets.

We downloaded all genomes from NCBI in July 2014 and selected any genome whose genus was found in the training sets of all methods, but the species was not found in any of the training sets of the marker-based methods. We then excluded the species of the selected genomes from the training sets of the composition-based methods. This resulted in a set of genomes that were novel to all methods at the species level, but not novel to any of the methods at the genus level. In total, 100 genomes met this criterion.

We used MetaSim to generate four datasets containing all 100 genomes, each dataset varying the complexity of the profile (LC versus HC) and the sequencing model used to generate the reads (short Illumina reads versus long 454 reads). We categorize these datasets as ‘easy’ novel genome datasets as there were no indels (datasets with short Illumina reads) or low amounts of indels (5% indel error per base for the datasets with long 454 reads). In addition, we generated ‘hard’ novel genome datasets by modifying the ‘easy’ novel genome datasets to include additional indel errors (12% insertion and 2% deletion error rate per base).

3.1.8 Computing profiles. MetaPhlAn, mOTU, MetaPhyler and TIPP output an abundance profile from a set of query sequences; abundance profiles for NBC and PhymmBL were estimated using the relative abundance of the query sequence.

3.1.9 Performance evaluation. We compute the Hellinger distance (Rao, 1995) between the estimated abundance profile and the true abundance profile. Let $C_i$ be the set of clades found in the true profile and the estimated profile for the taxonomic level $l$, $R_x$ be the abundance of clade $x$ for the true profile, and $E_x$ be the abundance of clade $x$ for the estimated profile. Then $H_l$ (the Hellinger distance for taxonomic level $l$) is:

$$H_l = \sqrt{\sum_{x \in C_l} (\sqrt{R_x} - \sqrt{E_x})^2}$$

$H_l$ ranges from 0 (if the profiles match exactly) to 1 (if the profiles share no taxa in common and the profiles have no unclassified clades). Note that if a method estimates a portion of the clade as unclassified, that portion is not included in the $H_l$ calculation (i.e. $E_{\text{unclassified}}$ is omitted from the formula). Finally, the MetaPhlAn HC and LC datasets consist of two and eight replicates; for these datasets, we report the average $H_l$ of the replicates.

3.1.10 Computational resources. The majority of experiments were run on the homogeneous Stampede cluster at the Texas Advanced Computing Center (TACC). Each method was run on a dedicated node with 16 cores with 32 GB of memory and given 48 h to complete on TACC (maximum allotted time that a job can run on Stampede). Methods that could not complete within the 48-h limit were also run on a heterogeneous Condor computing cluster and homogeneous Lonestar cluster.

4 RESULTS

Experiment 1: Easy datasets. Our first experiment explored datasets with low error rates coming from known genomes. On the long read easy datasets, mOTU had generally poor performance (Supplementary Table S2), and also failed to run on some of these datasets (terminated with an error message, see Supplementary Section S1.6); we, therefore, omit mOTU from the long read results.

Figure 2a shows the average Hellinger distances on the easy datasets; figures and tables for individual datasets can be found in Supplementary Section S1.1. Results on the long read datasets (left subfigure) show that TIPP had the best overall accuracy, followed by NBC. MetaPhlAn was in third place in terms of overall performance (though Metaphyler was more accurate than MetaPhlAn at the class level). Finally, PhymmBL was the
least accurate on these data. On the short read datasets (right subfigure), results were somewhat different. First, while mOTU had poor performance on the long read datasets, mOTU had the best overall accuracy of all methods on the short read datasets. At the species level, MetaPhlAn tied with mOTU for first place, followed by TIPP and then NBC. At the genus level, mOTU had the best accuracy, but TIPP and MetaPhlAn were very close. Results at the family level through class levels showed TIPP and mOTU tying for first, with MetaPhlAn in third place. Finally, at the phylum level, TIPP, MetaPhlAn, MetaPhyler and mOTU tied for first, with much better accuracy than NBC or PhymmBL.

Because MetaPhlAn was trained on its datasets, we removed MetaPhlAn’s datasets and re-evaluated performance (Fig. 2b). On this reduced set, MetaPhlAn tied for first at the species level (slightly less accurate than mOTU, but equal to TIPP), but was much less accurate than TIPP at all other levels. TIPP had the best overall performance on these data, though it tied for first with mOTU at the genus level, and tied for first with PhymmBL at the phylum level.

Experiment 2: Datasets with known genomes and high rates of indel errors. Figure 3 shows the average performance of different methods on datasets with known genomes and simulated with high rates of indel errors. We omit mOTU from the results because it failed to complete on any of these datasets. We omit MetaPhlAn, MetaPhyler, and NBC from the long read datasets as they could not classify any reads from at least one of the following datasets: WebCarma (MetaPhlan), FACS HC (NBC and MetaPhyler), and FAMeS LC (MetaPhyler). See Section S1.2 for results on individual datasets.

On the easy long read novel genome datasets (top left), TIPP had the best accuracy at all but the species level, where MetaPhyler was the best. The next best method was MetaPhlAn, and the remaining methods had comparable performances to each other (though NBC was slightly less accurate than the others at all levels). On the high indel long read novel genome datasets (bottom left), TIPP again had the best accuracy at all but the species level, where MetaPhyler was best. But on these more difficult long read datasets, MetaPhyler had very poor results at the remaining levels, largely because it
During the taxonomic profiling experiment. One of the important observations in this study is that the marker-based methods (TIPP, MetaPhlAn, mOTU and MetaPhyler) usually gave more accurate abundance profiles than the composition-based methods (NBC and PhymmBL), even though marker-based methods estimate abundance profiles by characterizing only those reads that map to their selected marker genes. These results show that accurate profiles can be obtained by taxonomic characterization of only a fraction of the query sequences. However, the choice of marker genes, and the technique used to bin the reads to the markers, has an impact on the resultant abundance profile. Thus, future research should investigate whether improved performance can be obtained using a different set of marker genes and different techniques to map reads to the marker genes. Also, our abundance profile estimates were based on combining all reads for all markers into one set, and using the distribution estimated for that set; more sophisticated techniques could be used to combine distributions estimated for each marker.

ACKNOWLEDGEMENTS

The authors thank Sean Eddy for his help with the derivations of the probability that a query sequence is generated by one HMM in a set of HMMs. The authors also thank the Huttenhower lab for sharing the synthetic data with us for the taxonomic profiling experiment.

Funding: This research was partially supported by a Guggenheim Foundation Fellowship to T.W.; National Science Foundation grants DBI-1062335, DBI-1461364 and DEB 0733029 to T.W.; an HHMI International Predoctoral Fellowship to S.M.; the iPlant Collaborative U.S. National Science Foundation grant DBI-1265383 (via TACC) to N.N.; and National Institutes of Health grant R01-AI-100947 to M.P. Some of this work was performed while T.W. was a program director working for the US National Science Foundation, and supported by the IR/D program.

REFERENCES


